

JackBytes

A simple, free, opensource server for relaying sound information

Robert Atwood

¹Department of Jack Bytes
www.jackbytes.net

Piksel07

Outline

- 1 Implementation
 - Dependencies
 - Interface
- 2 Examples
- 3 Demonstration

Philosophical, Political, ...

- Certain people, doing nice work in reactive audio-visual performances, were using *non-free* packages for enabling their application to react to the sound's loudness and pitch
- Running on *You-know-what* Operating system!
- I could *hissss*, and curse them out, I suppose, when in the context of *free software* event...
- ... Or, I could fix it myself!

Philosophical, Political, ...

- Certain people, doing nice work in reactive audio-visual performances, were using *non-free* packages for enabling their application to react to the sound's loudness and pitch
- Running on *You-know-what* Operating system!
- I could *hissss*, and curse them out, I suppose, when in the context of *free software* event...
- ... Or, I could fix it myself!

Visuals reacting to Sound

- Loudness
 - Use the raw sound data values added up and averaged (*root mean square*)
- Pitch
 - Obtain the *frequency spectrum* via an FFT
 - Either a textbook solution (fixed sample number) or FFTW library (arbitrary sample number)
 - actually I am not sure about the “strict” free-software-status of a textbook solution!

Jack

- Flexible
 - You can select the sound device, or another application
 - You can configure different numbers of ports
(not yet in JackBytes)
- Popular
- Easy to use ... for a C programmer...
- I use it ...
- ... also it would be easy to get input from PortAudio instead!
Hi, Ross!

Bytes

- For most reactive visuals, more than 8-bit sound resolution is not really necessary
- Also, it simplifies the transfer of data
- I need not worry about network-byte-order and so on
- The client language may not have a simple method for getting other datatypes from the stream
e.g. in *Proce55ing*, which is java based, I don't know an obvious way to change an array of bytes that represents floats into floats besides brute-force
- Pitch resolution depends upon the size of the transformed packet anyways

JackBytes

- Makes the data available when it is not feasible for the application to access Jack and/or access the frequency analysis itself
- Does not do more than is necessary
 - (*Keep it Simple, Stupid!*)
 - ... or should that be ...
 - (*Keep it Simple AND Stupid!*)
- The data is available via a TCP port
- Does not need to be the local host
- I hope you are not too worried about network latency
- Simple single-byte commands (only 6 defined including testing)

- 1 **Implementation**
 - Dependencies
 - Interface
- 2 Examples
- 3 Demonstration

Dependencies

Certain software is required for building and using JackBytes.

- JACK audio connection kit.
GPL, LGPL
- FFTW Fastest Fourier Transform in the West.
GPL
- CMake build system (optional, so far).
Berkely-style
- The usual suspects
(C compiler, make, standard libraries, TCP sockets)

The client/server interface

The client must open its end of the TCP socket, and then use the following single byte commands:

- **D, T, Z, n, Q, and q**
- D: Request some raw data
- T: Request some frequency data
- Z: Request some test data
- n: Set a new data set size (followed by 2 bytes)
- Q: Quit the connection
- q: Quit the program

Any other command causes the currently selected kind of information (raw, transformed, or test) to be sent.

JackBytes c++ Client Interface

The class encapsulates some commmands for it more convenience, e.g. getData sends the 'D' command then receives the bytes and puts them in the default buffer.

```

/* client opening methods */
void jb_open(int portno, int rate, const char * const hostname);
void jb_close();
/*****/
/* initialize methods */
void start (u_int32_t slength);
void set_array_size (u_int32_t slength);

/* methods to get some data */
void getSpectrum();
void getData();
void sendCommand(byte cmd);
void getBytes();
void getBytes(byte * outdat);
float getRMS();
void getRMS(byte * outdat);
void setSize (u_int16_t newszie);

```

JackBytes Processing Client Interface

```

/* JackBytes Processing Client Class */
/* THIS IS VERSION 06 */
Jackbytes(PApplet ap)
    /* with no arguments, default to port 9999 at localhost and cd soundrate */
    /* The actual sampling rate of the data is determined by JACK */
Jackbytes(PApplet ap,int portno, int rate)
    /* default, open localhost */
Jackbytes(PApplet ap,int portno, int rate, String hostname)
    /* client opening methods */
void jb_open(int portno, int rate, String hostname)
void jb_close()
void start (int slength)
void set_array_size(int slength)
void setSize(short newszie)
    /* methods to get some data */
void getSpectrum() /* request frequency data */
void getData()     /* request sound data */
void getBytes(byte [] outdat) /* used by above to fill array */
  
```

- 1 Implementation
 - Dependencies
 - Interface
- 2 **Examples**
- 3 Demonstration

C++ trivial application example

The *getbytes* program just requests data and prints it to the console.

```
jb = new Jackbytes(pnum,44100,argv[1]);
jb->start(nrecv);

while (qflag == 0) { /* loop until quit */

printf ("Please enter the keystroke: \n");
fgets (buffer, 255, stdin);

switch (buffer[0]){
case 'D':
    jb->getData();
break;
case 'T':
```

... *and so on*

Show the Example

Processing Example

Order of starting the jacbytes client:

- create with 'this' as creation argument
- start, giving the array length as argument

```
jackbytes = new Jackbytes(this);
jackbytes.start(1024);
```

- Later on, call getSpectrum() then use the data:

```
jackbytes.getSpectrum();
for (i=1;i<1024;i++){
  stroke(0,255,0);
  if (jackbytes.spectrum[i]>1){
    ellipse(i*width,128-jackbytes.spectrum[i],
           jackbytes.spectrum[i],
           jackbytes.spectrum[i]);
  }
}
```

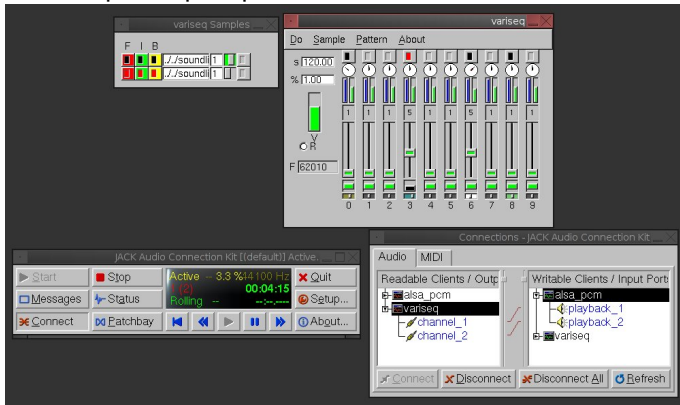
Show the Example

A Little Performance

My *Variseq* sequencer application may be connected via **JACK** and **JackBytes** to my *Breakimage* application that uses **OpenGL** to move bits of image around the screen.

Variseq

A sample step sequencer with the features I wanted.



Input and output can be connected to **JACK**

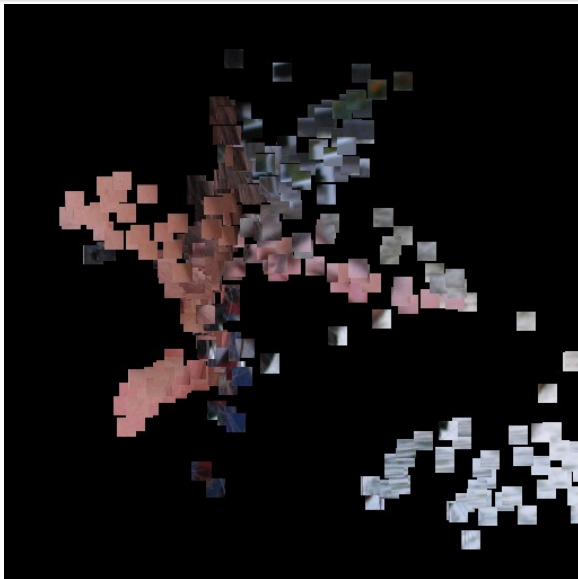
Breakimage

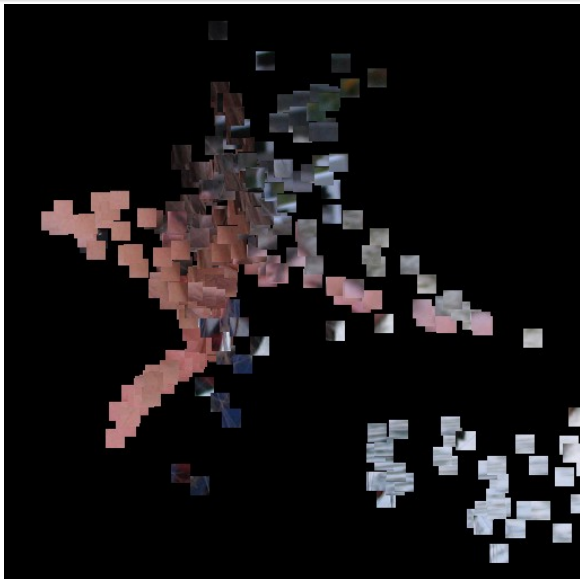
The force on each piece depends on:

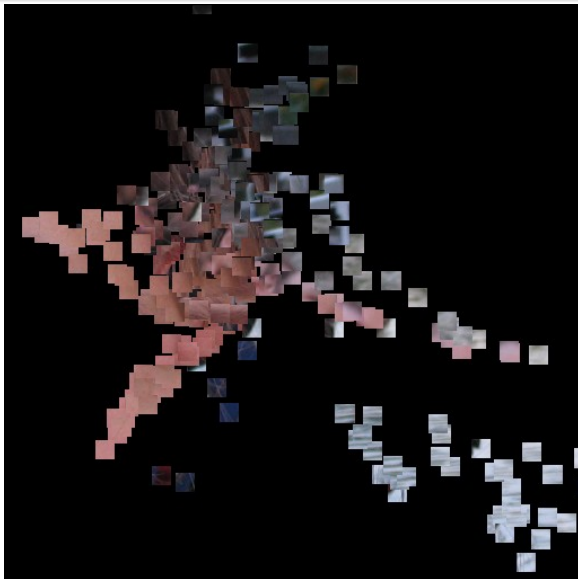
- The difference in average colour between the tiles.
This is defined as the vector distance in (r,g,b) space after averaging the red, green and blue components within each tile.
- The distance between the tiles.
- The loudness of the sound.

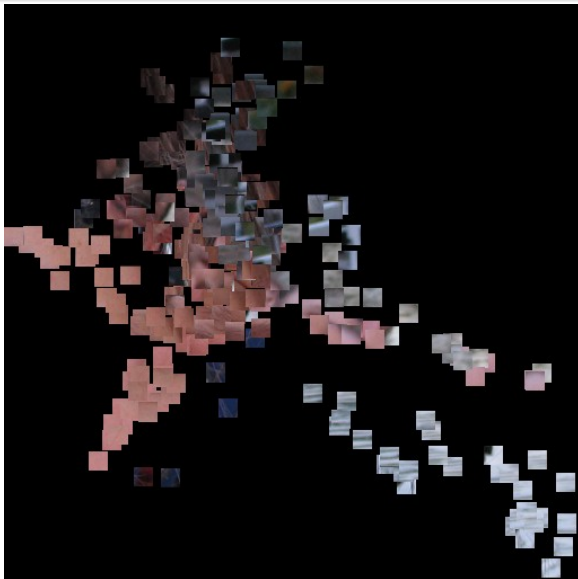
Then, the coordinates of each tile corner are altered depending upon:

- The average loudness of the frequency component corresponding to the average brightness of the tile.









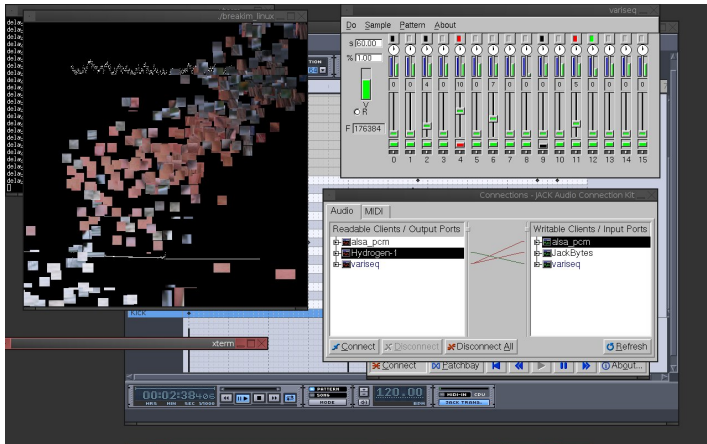
Problems

The whole process is still a little clumsy to get started.

- Start JACK
- Start the sound application (variseq)
- Start JackBytes
- Start the visual application (breakimage, processing)
- Connect the correct ports within JACK
- **Ensure JACK is ROLLING**
This seems easy to forget, does nobody else use the 'rolling' feature of JACK?
- Start playing!

- 1 Implementation
 - Dependencies
 - Interface
- 2 Examples
- 3 Demonstration

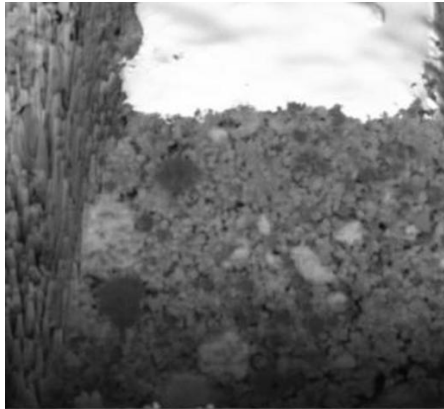
A Little Performance



Hydrogen and Variseq sending sound data to Breakimage via
JackBytes

Show the Example

A Little Performance

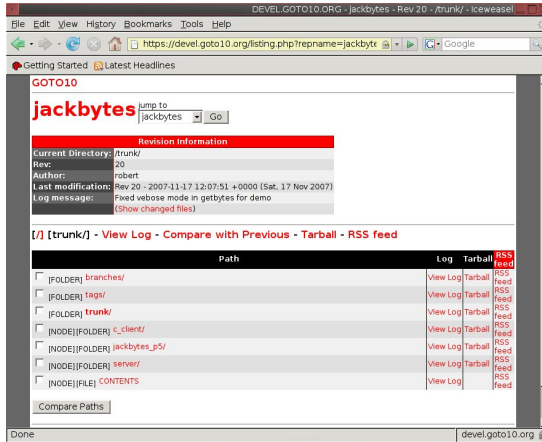


Greyscale image to illustrate frequency-based glitching

Directions

Some future directions


- Improved frequency analysis e.g. Proper data windowing
- More than 1 port ?
- A **precompiled *Processing* client library** is under development by Oli Laurelle of goto10 .



The **source code repository** is hosted at
devel.goto10.org

Thank You!

- L^AT_EX Beamer graphical styles provided by the L^AT_EX Beamer package, by Till Tantau and friends.
- Screen capture of goto10 website by Goto10.org members.

 © 2007 R. C. Atwood. This work is licensed under the Creative Commons Attribution-Share Alike 2.0 UK: England & Wales License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.0/uk/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.